REMARKS

In the Office Action, the Examiner rejected claims 1-14, 16-32, and 34-50. By this paper, claim 48 is cancelled and claims 1, 13, 16, 23, 28, 45, 46, 49, and 50 are amended for clarification of certain features set forth in the recited subject matter. Claims 15 and 33 remain cancelled. No new matter has been added. As such, claims 1-14, 16-32, 34-47, and 49-50 are currently pending and believed to be in condition for allowance. Applicant respectfully requests reconsideration of the Application in view of the following remarks.

Preliminary Remarks

In a Pre-appeal Brief Request for Review filed by Applicant on May 7, 2007, Applicant presented to the Panel several arguments stating why the Examiner's primary reference, Inagami et al., U.S. Patent No. 4,881,168 (hereinafter "the Inagami reference"), fails to disclose several features recited by the pending claims. *See* Pre-appeal Brief Request for Review filed on May 7, 2007. Specifically, Applicant asserted that the Inagami reference (1) fails to show an embedded-DRAM as recited by the pending claims, and (2) that the use ***separate commands*** for row precharge, row deactivate, and loading operations, are *not inherently present* in the DRAM device taught by Inagami, as suggested by the Examiner. *See id.* at pages 2-5. Following a Pre-Brief Conference, the Panel instructed the Examiner to reopen prosecution of the present Application. *See* Notice of Panel Decision mailed July 2, 2007. Indeed, the Panel agreed with the Applicant that the Inagami reference is deficient for at least the reasons presented in the above-referenced Pre-appeal Brief Request for Review. *See id.* As such, pursuant to the Panel's decision, the instant Office Action was mailed on August 21, 2007.

As a preliminary matter, Applicant notes that throughout the prosecution of the present Application, the Examiner has been unable to find a single reference anticipating the recited subject matter. Instead, the Examiner has continued to maintain obviousness rejections under 35 U.S.C. § 103(a), relying on various combinations selected from four or more references, an overwhelming use of Official Notice in several previous communications, as well as assertions that certain features recited in the pending claims are *inherent* in the cited art. *See* Office Action mailed December 9, 2004; *see also* Final Office Action mailed July 25, 2005; *see also* Office Action mailed April 5, 2006; *see also* Final Office Action mailed January 30, 2007. Following the Panel Decision, in the instant Office Action, the Examiner again maintained that the primary reference, Inagami, fails to disclose several elements of the pending claims. However, like previous Office Actions, the Examiner has cited a combination of additional references, each additional reference being used solely to address a *single* element which the Examiner is unable to find in the Inagami reference. For example, in the instant Office Action, the Examiner cites two *new* references, Luk et al., U.S. Patent No. 5,883,814 (hereafter referred to as "the Luk reference"), and Farmwald et al., U.S. Patent No. 5,243,703 (hereafter referred to as "the Farmwald reference"), each used solely to address one of the above referenced deficiencies of the Inagami reference. As illustrated below, the Luk and Farmwald references are cited along with previously cited references in an attempt to remedy the numerous deficiencies of the Inagami reference:

> (1) Luk is cited solely to address an embedded DRAM which is not present in Inagami;
>
> (2) Farmwald is cited solely to address activation/deactivation commands which are not present in Inagami;
>
> (3) Parady, U.S. Patent No. 5,933,627 (hereafter referred to as "the Parady reference") is cited solely to address active and inactive states of register files which are not present in Inagami; and
>
> (4) Bissett et al., U.S. Patent No. 5,896,523 (hereafter referred to as "the Bissett reference") is cited solely to address a command for transferring data between a DRAM and an inactive register file which is not present in Inagami.

Regardless, Applicant respectfully asserts that *any* possible combination of the numerous cited references fails to disclose at least one element recited by the pending claims, as will be discussed below.

The Examiner's burden of proof.

Applicant respectfully reiterates that, in accordance with established legal precedent, the burden of establishing a *prima facie* case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (B.P.A.I. 1979). To establish a *prima facie* case, the Examiner must show, among other things, that the combination includes *all* of the claimed elements. *Ex parte Clapp*, 227 U.S.P.Q. 972 (B.P.A.I. 1985). Moreover, in the recent case of *KSR International Co. v. Teleflex Inc.*, the U.S. Supreme Court noted that "a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art." *KSR International Co. v. Teleflex Inc.*, No. 04-1350 at 14 (April 30, 2007). The *KSR* court acknowledged that "because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known," that it is important that the Examiner identify a reason that would have prompted a person of ordinary skill in the relevant art to combined the elements in the exact manner that the new invention does. *Id.* at 14-15. In other words, for each reference that the Examiner attempts to add to the combination of cited art, the Examiner must not only demonstrate that the combination includes *all* the claimed elements, but also provide some objective evidence as to why a person of ordinary skill in the art would be prompted to combine the cited references the same as the recited subject matter of the invention.

With this controlling case law in mind, due to the numerous references cited in combination throughout the prosecution of this Application, Applicant reminds the Examiner that the burden of proof for combining *each reference* must be established. However, the Examiner has failed to provide objective evidence, as required by the *KSR* court, oftentimes providing only *mere conclusory* statements that it would have been obvious to combine these additional references with previously cited references without providing any objective evidence of record. Indeed, the factual inquiry whether to combine references must be *thorough and searching*, and that it must be based on *objective evidence of record*. *In re Lee,* 61 U.S.P.Q.2d 1430, 1436 (Fed. Cir. 2002). (Emphasis added).

With this in mind, it should be noted, however, that Applicant does not believe any combination of the cited references in the instant Office Action anticipates or renders obvious any of the pending claims, as will be discussed below. However, even assuming hypothetically that the combination of the cited references *did* disclose all the claimed elements, Applicant also does not believe the Examiner has sufficiently provided objective evidence demonstrating that one skilled in the art would have been prompted to combine the references in the same manner.

The cited references, alone or in combination, do not disclose all the elements recited in the pending claims.

Applicant respectfully notes that the present invention is directed generally towards an embedded-DRAM processor architecture utilizing an intelligent caching scheme which operates based on an instruction set. *See* Application, Abstract. The instruction set includes several commands which, among other things, control the movement of data throughout the system. *See id.* As such, the techniques presented in the present Application minimize the amount of instruction cache needed to execute programs at full speed from a DRAM-oriented program memory. *See id.* Furthermore, although Applicant believes all the pending claims to be allowable as previously presented, Applicant has further amended the claims by this Response for clarification of certain features not present in any of the cited references. Accordingly, in view of these amendments, Applicant believes that the combination of the cited references, as suggested by the Examiner, *clearly* does not anticipate or render obvious the pending claims for at least the following reasons:

(1) The cited references, alone or in combination, do not teach or suggest dual-port register files having first access port operative to parallely transfer contents of a selected register file between a DRAM row and at least a second access port operative to transfer data between the selected file register and a functional unit;

(2) The cited references, alone or in combination, do not teach or suggest register files capable of loading or storing either an entire

row (or selected columns of an entire row) of a DRAM array either in a single operation or in response to a single latch signal, nor do the cited references teach or suggest a command for doing the same;

(3) The cited references, alone or in combination, do not teach or suggest speculatively precharging a row based at least partially on historical program execution data;

(4) The cited references, alone or in combination, do not teach or suggest a command to deactivate a row pointed to by a row address register after the row has been precharged; and

(5) The cited references, alone or in combination, do not teach or suggest register files capable of being placed into an active and an inactive state, nor do the cited references teach or suggest a command for doing the same.

The foregoing deficiencies will by fully discussed in the subsequent paragraphs.

Incomplete Examination: Request Examination of Claim 51 in a future <u>non-final</u> Office Action.

Applicant respectfully notes that <u>claim 51</u> (added by Applicant's response to the Office Action mailed on January 30, 2007) was not addressed by the Examiner in the present Office Action. As such, Applicant assumes that claim 51 contains allowable subject matter. Therefore, Applicant requests that the Examiner allow claim 51 (along with claims 1-50), or else explicitly set forth in a future <u>non-final</u> office action the reason or reasons why claim 51 would not be allowable.

<u>Objection to the Specification</u>

In the Office Action, the Examiner objected to the Specification. Specifically, the Examiner stated that the title of the invention is not descriptive and requested that the Applicant supply a new title that is indicative of the invention to which the claims are directed. *See* Office Action, page 2, paragraph 3. Applicant respectfully traverses the Examiner's objection. First, Applicant notes that the title of the present Application was previously amended to "PROGRAM

CONTROLLED EMBEDDED-DRAM-DSP ARCHITECTURE AND METHODS" in order to address a previous objection by the Examiner. *See* Office Action mailed July 25, 2005, page 3, paragraphs 4-5 (the Examiner objecting to the title as not being descriptive of the invention per M.P.E.P. § 606.01); *see also* Response to Office Action mailed July 25, 2005, pages 17-18. Moreover, Applicant's previous amendment to the title appeared to adequately address the Examiner's previous objection, as the Examiner made no further objections regarding the title in subsequent Office Actions mailed on April 5, 2006 and January 30, 2007. Therefore, it is unclear to the Applicant on what new grounds, if any, the Examiner is objecting to the title in the instant Office Action.

Furthermore, in the present objection, the Examiner has provided no basis for his assertion that the title, as previously presented, fails to be descriptive. Section 606 of the M.P.E.P. states that "[t]he title of the invention may not exceed 500 characters in length and must be as *short* and *specific* as possible." M.P.E.P. § 606 (citing 37 C.F.R. § 1.72). The present invention is directed to an embedded DRAM processor architecture utilizing a data assembly approach for intelligent caching, thereby reducing the amount of instruction cache needed to execute programs at full speed from a DRAM-oriented program memory as compared to the prior art. Applicant cannot understand how "PROGRAM CONTROLLED EMBEDDED-DRAM-DSP ARCHITECTURE AND METHODS" can be considered non-specific or non-descriptive under Section 606. Moreover, Applicant believes the previously presented title to be completely descriptive of the broadest generic embodiment of the invention presented herein.

Nevertheless, although Applicant traverses the Examiner's objection, by this paper, Applicant has further amended the title of the Application as follows:

> PROGRAM CONTROLLED EMBEDDED-DRAM-DSP HAVING
> IMPROVED INSTRUCTION SET ARCHITECTURE

As such, Applicant believes the present revision is *as descriptive as possible* without unduly limiting or affecting the genericism of the title. Therefore, Applicant respectfully requests that the Examiner withdraw the objection to the title. However, if the Examiner finds that the present attempt at revision of the title is still not suitable, Applicant respectfully requests that the Examiner: (i) provide a specific reasoning or support for such assertion, and (ii) supply a title which the Examiner believes to be generic to all claims and sufficiently descriptive.

Claim Rejections Under 35 U.S.C. § 103(a)

In the Office Action, the Examiner rejected claims 1, 6-10, 12-14, and 46-48 under 35 U.S.C. § 103(a) as being unpatentable over the Inagami reference, in view of the Luk reference, and further in view of the Farmwald reference, and, in addition, Garcia, et al., U.S. Patent No. 4,827,476 (hereafter referred to as "the Garcia reference") is cited as extrinsic evidence which shows that DRAM is inaccessible during refresh cycles. The Examiner also rejected claims 23-25 under 35 U.S.C. § 103(a) as being unpatentable over the Inagami reference in view of the Luk reference; rejected claims 2-5 and 11 under 35 U.S.C. § 103(a) as being unpatentable over the Inagami reference in view of the Luk reference in view of the Farmwald reference and further view of the Parady reference; rejected claims 26 and 27 under 35 U.S.C. § 103(a) as being unpatentable over the Inagami reference in view of the Luk reference and further in view of the Parady reference; rejected claims 16-22, 28-32, 34-43, 45, and 49-50 under 35 U.S.C. § 103(a) as being unpatentable over the Inagami reference in view of the Luk reference, in view of the Parady reference, and further in view of the Bissett reference; and rejected claim 44 under 35 U.S.C. § 103(a) as being unpatentable over the Inagami reference in view of the Luk reference in view of the Parady reference in view of the Bissett reference and further in view of the Farmwald reference. Applicant respectfully traverses these rejections.

Legal Precedent

First, the pending claims must be given an interpretation that is reasonable and consistent with the *specification*. *See In re Prater*, 415 F.2d 1393, 1404-05, 162 U.S.P.Q. 541, 550-51 (C.C.P.A. 1969) (emphasis added); *see also In re Morris,* 127 F.3d 1048, 1054-55, 44

U.S.P.Q.2d 1023, 1027-28 (Fed. Cir. 1997); *see also* M.P.E.P. §§ 608.01(o) and 2111. Indeed, the specification is "the primary basis for construing the claims." *See Phillips v. AWH Corp.*, 415 F.3d 1303, 75 U.S.P.Q.2d 1321 (Fed. Cir. 2005) (*en banc*). One should rely *heavily* on the written description for guidance as to the meaning of the claims. *See id.* Moreover, interpretation of the claims must also be consistent with the interpretation that *one of ordinary skill in the art* would reach. *See In re Cortright*, 165 F.3d 1353, 1359, 49 U.S.P.Q.2d 1464, 1468 (Fed. Cir. 1999); M.P.E.P. § 2111.

Second, the burden of establishing a *prima facie* case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (B.P.A.I. 1979). To establish a *prima facie* case, the Examiner must show, among other things, that the combination includes *all* of the claimed elements. *Ex parte Clapp*, 227 U.S.P.Q. 972 (B.P.A.I. 1985). (Emphasis added). Furthermore, in addressing obviousness determinations under 35 U.S.C. § 103, the Supreme Court in *KSR International Co. v. Teleflex Inc.*, No. 04-1350 (April 30, 2007), reaffirmed many of its precedents relating to obviousness including its holding in *Graham v. John Deere Co.,* 383 U.S. 1 (1966). In *KSR*, the Court also reaffirmed that "a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art." *Id.* at 14. In this regard, the *KSR* court stated that "it can be important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does ... because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known." *Id.* at 14-15. In *KSR*, the court noted that the demonstration of a teaching, suggestion, or motivation to combine provides a "helpful insight" in determining whether claimed subject matter is obvious. *KSR, slip op.* at 14.

Furthermore, the *KSR* court did not diminish the requirement for objective evidence of obviousness. *Id.* at 14 ("To facilitate review, this analysis should be made explicit. See *In re Kahn,* 441 F.3d 977, 988 (CA Fed. 2006) ("[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness"). As our precedents make clear, however, the analysis need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ."); s*ee also, In re Lee,* 61 U.S.P.Q.2d 1430, 1436 (Fed. Cir. 2002) (holding that the factual inquiry whether to combine references must be thorough and searching, and that it must be based on *objective evidence of record*).

As noted above, the cited references, alone or in combination, fail to disclose several of the elements recited in the pending claims. The Applicant's remarks in the subsequent paragraphs regarding these deficiencies should be considered with the foregoing legal precedent in mind. Moreover, the Applicant has provided the Examiner with an itemized list of several of the deficiencies, as shown above. It should be understood, however, that the deficiencies listed herein are not exclusive, and that there may be further additional deficiencies in the cited references.

Deficiency #1: The cited references, alone or in combination, do not teach or suggest dual port register files having a first port operative to parallely transfer contents of a selected register file between a DRAM row and at least a second port operative to transfer data between the selected file register and functional unit.

With regard to the first deficiency of the cited references noted above, Applicant notes that this subject matter is recited at least by independent claims 23, 45, and 50. For example, independent claims 23, 45, and 50 each recite, generally, first and second dual port register files comprising *both* a first access port operative to parallely transfer contents of one register between a DRAM row *and* a second access port operative to transfer data between a selected one of said registers and a functional unit. For example, the dual port feature of the register files is clearly illustrated in Fig. 2 of the present Application. Referring to Fig. 2, an embodiment of the present invention clearly shows a dual port register file 112, wherein the dual port feature is provided by a series of switches 204 coupled to each of the registers making up the register file. *See* Application, Fig. 2. Each of the switches 204 includes a first access port coupled to a parallel load/store channel for transferring data to a DRAM array. *See* Application, page 28, lines 4-20. Furthermore, each switch 204 also includes a second access port coupling the register file to a functional unit 128 (through selector switch 206). *See id.* By this second access port, the selector switch 206 selectively couples the registers of the register file 112 either to the functional units 128 or to the data assembly unit 122. *See id.* In other words, the register files, as recited by claims 23, 45, and 50, include two *separate* and *distinct* access ports: (1) a *first access port* that operatively transfers data to or from the register file and a row in the DRAM array, and (2) a *second access port* that transfers data from the register file to a functional unit, such that the dual port switches 204 are connected to both the DRAM array and the functional unit in a *parallel* manner via the *separate* access ports.

Applicant respectfully notes that the above described subject matter appears to be entirely absent from the Inagami reference. In the present rejection, the Examiner, correlating the dual port feature of the register files as being provided by switches 120-123, the DRAM row to the main storage structure 1, and the functional unit to the load/store sub-pipe 2, stated that Inagami discloses a dual port register file having a first access port operative to parallely transfer data between the register file and a row of DRAM, as well as a second access port operative to transfer data between the register and a functional unit. *See* Office Action, pages 16, 41, and 48; *see also* Inagami, Fig. 1. However, after reviewing the passages and figures cited by the Examiner, Applicant does not believe the Inagami reference teaches or suggests dual port register files having *both* a first access port operative to parallely transfer data from a register file (i.e., register files VR0-VR7) to a DRAM (i.e., main storage 1) *and* a second access port operative to transfer data between the register file and a functional unit (i.e., load/store sub-pipe 2) as suggested by the Examiner. Furthermore, Applicant can find no basis in any other passage of the Inagami reference supporting the Examiner's position.

To the contrary, the Inagami reference merely discloses that a register file, functional unit, and a row in a DRAM array are coupled in *series* via a *single* access port, not a *dual access port* as recited by claims 23, 45, and 50. For example, referring to Figure 1 of Inagami, a DRAM array (main storage 1) and a functional unit (load/store sub-pipe block 2) are illustrated and coupled via input/output connection lines 220-223 and 230-233. *See* Inagami, Fig. 1. Furthermore, the *only* connections between each of switches 120-123 (which the Examiner has correlated to each of the dual port switches 204 of the present invention) and the main storage 1 and load/store sub-pipe block 2 are respective *single connection lines* for each switch 120-123, designated in Fig. 1 of Inagami by reference numerals 240-243. Thus, even if the switches 120-123, which are associated with what the Examiner has identified as register files (VR0-VR7), can be properly correlated with the dual port switches 204 illustrated in Fig. 2 of the present Application, the switches 120-123 *clearly* do not show *both* a first access port for transferring data between the register file and a functional unit (load/store sub-pipe 2), *and* a second access port for transferring data between the register file and a row of the DRAM array (main storage 1). Rather it appears that any data transferred to or from the DRAM array (main storage 1) to the register files (VR0-VR7) must pass through the *same* access port connecting the functional unit (load/store sub-pipe 2) to the register files. *See id.* In other words, the Inagami reference only discloses a single access port. It does not, as the Examiner suggests, disclose separate access port or ports which would provide a direct connection between the main storage 1 and the switches 120-123 for the transfer of data.

Moreover, Applicant notes that the Examiner seems to be aware of the deficiency of a second access port in the Inagami reference. For instance, the Examiner's *only* reasoning that a second access port exists for transferring data is that "if transfers occur between DRAM and registers…then a port *must* couple registers to DRAM." Office Action, page 41. (Emphasis added). At best, this statement appears to be a modest attempt at an assertion that a second access port is *inherently* taught by the Inagami reference. However, if the Examiner relies on a theory of inherency, the extrinsic evidence must make clear that the missing descriptive matter is *necessarily* present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. *In re Robertson,* 169 F.3d 743, 49 U.S.P.Q.2d 1949 (Fed. Cir. 1999) (Emphasis Added). The mere fact that a certain thing *may* result from a given set of circumstances is not sufficient. *Id.* Thus, in view of the controlling case law, any such assertion that a second access port must exist *merely* because data transfer may occur between the register files (VR0-VR7) and the DRAM array (main storage 1) must necessarily fail when examined in view of Inagami, for Inagami *clearly* shows that data transfer between the register files (VR0-VR7) and one or more rows of the DRAM array (main storage 1) may be accomplished through the *same* access port which allows transfer of data between the register files (VR0-VR7) and the functional unit (load/store sub-pipe 2).

Indeed, there is *clearly* no second access port, as recited by claims 23, 45, and 50, taught either explicitly, implicitly, or inherently by Inagami and, therefore, Applicant asserts that the Inagami reference *clearly* fails to teach or even suggest dual port register files having *both* a first access port operative to parallely transfer data between a register file and a row of a DRAM array *and* a second access port operative to transfer data between the register file and a functional unit. Moreover, because none of the additional references cited by the Examiner, either alone or in combination, obviate the deficiencies of the Inagami reference, Applicant respectfully asserts that the Examiner has failed to establish a *prima facie* case of obviousness with regard to this subject matter. For at least this reason, Applicant respectfully requests withdrawal of the rejection under 35 U.S.C. § 103(a) of independent claims 23, 45, and 50, as well as those claims which depend thereon.

Deficiency #2: The cited references, alone or in combination, do not teach or suggest register files capable of loading or storing either an entire row or selected columns of an entire row of a DRAM array either in a single operation or in response to a single latch signal, nor does the prior art teach or suggest a command for doing the same.

With regard to the second deficiency of the cited references noted above, Applicant notes that this subject matter can be found at least in independent claims 1, 13, 16, 23, 28, 46, 49 and 50. In rejecting the claims with regard to this subject matter, the Examiner again relies on the Inagami reference. Specifically, the Examiner stated that the Inagami reference discloses register files (VMR0-VMR7 and VR0-VR7) which may be loaded or stored in response to a single load or store signal. *See* Office Action, pages 4, 9, 11, 16, 31, 44, 47. The Examiner further stated that the Inagami reference discloses a command for loading either selected elements of a row or an entire row of a DRAM array into the register files. *See id.* at pages 4, 10, 12, 17, 33, 45, 49.

While Applicant does not necessarily agree with the Examiner's position, Applicant has amended claims 1, 13, 16, 23, 28, 26, 49, and 50 in order to clarify certain features recited in the claims. For example, independent claims 1, 13, 16, 23, and 50, as amended, each recite, generally, one or more register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal. Independent claims 28 and 49, as amended, each recite, generally, that parallel transfer of data between an entire row of a DRAM array and a register file occurs in a single operation. Further, independent claims 1, 13, 16, 23, 28, 46, 49, and 50, as amended, each recite, generally, commands for loading an entire row or selected columns of a row of the DRAM (the selection based on a bit mask) into a selected register file in a single operation. In view of the claim amendments, Applicant respectfully asserts that the Inagami reference fails to teach or suggest register files capable of loading or storing an *entire row* of a DRAM array in response to a single latch signal, and also fails to teach or suggest register files capable of parallely transferring data between an *entire row* of a DRAM array in a single operation. Applicant also asserts that the Inagami reference fails to teach or suggest a command for loading either an *entire row* or selected columns of the *entire row* of a DRAM array into a register file in a single operation.

In contrast to the recited subject matter, the Inagami reference discloses a DRAM system in which at least four load operations (and thus four latch signals, one latch signal for each load operation) are required to load an entire row of a DRAM into a register file. Particularly, Applicant notes that in the instant rejection, the Examiner repeatedly refers to Figs. 4 and 5 of the Inagami reference. However, Figs. 4 and 5 *clearly* show that a register file (i.e., vector register 23) is divided into four storage locations, each storage location including four elements, which the Examiner has correlated to data registers within the vector register file 23. By way of example, the first storage location includes elements a0-a3, the second storage location includes elements a4-a7, and so forth. Therefore, in order for a row of the DRAM (i.e., vector data in main storage 21) to be loaded into the vector register file 23, the first four elements are loaded into the first storage location, the second four elements are loaded into the second storage location, and so forth. *See* Inagami, Fig. 4.

Furthermore, each of the load operations is *separate* and *distinct* from the others. For example, Inagami discloses that in a first load operation, vector elements a0 to a3 are compared with a vector mask 22 and are loaded accordingly by the load/store sub-pipes 2-0 to 2-3. *See id.* at col. 6, lines 48-53. Further, in a second load operation, vector elements a4 to a7 are compared with the vector mask 22 and loaded accordingly in a similar manner as the first load operation. *See id.* at col. 6, lines 53-63 (note, however, that only a4-a6 are loaded due to the vector mask). A third load operation follows, loading the next four elements, along with a fourth load operation for loading the last four elements. *See id.* at col. 6, line 63 to col. 6, lines 8. Fig. 5 of Inagami shows a similar technique for executing a series of store operations (essentially the reverse process of the load operations described in Fig. 4), wherein data from the vector register 32 is stored into the main storage (designated by vector data structure 12) by *separate* first, second, third, and fourth load store operations. *See id.* at col. 7, lines 14-36; Fig. 5. Indeed, the Inagami reference *clearly* shows that at least *four* different operations are required for loading *and* for storing a vector register 23, which the Examiner has correlated to a register file. In other words, contrary to the recitations of claims 1, 13, 16, 23, 28, 46, 49, and 50, which *clearly* state that a register file is capable of being loaded and/or stored in a single operation, the DRAM device taught by Inagami requires a *plurality* of separate and distinct operations for loading or storing a register file. Also, because Inagami requires at least four load or store operations, there must be a separate latch signal associated with *each* of the four operations for transferring the groups (i.e., elements a0-a3) of data. Therefore, the Inagami reference also fails to teach that an entire row of the DRAM can be loaded into a register file in response to a single latch signal. Moreover, even if it can be argued that the multiple load operations taught by Inagami occur in parallel (at the same time), loading a row of data into the vector register files still requires *four separate* load operations. As such, the Inagami reference *clearly* fails to disclose register files capable of loading or storing an entire row of a DRAM array in a single operation or via a single latch signal, as recited by the pending claims.

Keeping this distinction in mind, Applicant notes that the above-discussed passages of the Inagami reference describing the requisite need for multiple load and store operations in a DRAM device is specifically describing a "load expansion instruction," which the Examiner has correlated to a load command, as recited in the pending claims. *See* Inagami, col. 6 lines 32-36 (stating that the description of the load/store operations with regard to Fig. 4 is an explanation of the "load expansion instruction"). As noted above, claims 1, 13, 16, 23, 28, 46, 49, and 50, recite, generally, a load command for loading an *entire row* or selected columns of the *entire row* (the selection based on a bit mask) of a DRAM array into a register file in a *single operation*. However, to the extent that the Inagami reference may disclose what could arguably be a load command, Inagami does *not* disclose that the command to load the register file with an entire row of the DRAM array is performed in a single operation. Rather, as Applicant has *clearly* pointed out above, the load expansion instruction that the Examiner relies on in reaching the present rejection would require *several separate* load operations (i.e., four load operations) in order to load an entire row of a DRAM array into a register file.

In view of the foregoing analysis, Applicant respectfully asserts that the Inagami reference fails to disclose the following: (1) register files capable of loading or storing an *entire row* of a DRAM array in response to a single latch signal; (2) parallel transfer of data between an *entire row* of a DRAM array and a register file occurring in a single operation; and (3) commands for loading an entire row or selected columns from the entire row of the DRAM (the selection based on a bit mask) into a selected register file in a single operation. Moreover, because none of the additional references cited by the Examiner, either alone or in combination, obviate the deficiencies of the Inagami reference, Applicant respectfully asserts that the Examiner has failed to establish a *prima facie* case of obviousness with regard to this subject matter. For at least this reason, Applicant respectfully requests withdrawal of the rejection under 35 U.S.C. § 103(a) of independent claims 1, 13, 16, 23, 28, 46, 49, and 50, as well as those claims which depend thereon.

Deficiency #3: The cited references, alone or in combination, do not teach or suggest speculatively precharging a row based upon historical program execution data.

With regard to the third deficiency of the cited references noted above, Applicant respectfully notes that this subject matter is recited at least by independent claim 46. For example, independent claim 46, as amended, recites, generally, speculatively precharging a row pointed to by at least one row address register based at least partially upon *historical program execution data* indicating a possible need to perform one or more load or store operations that would access said row. After careful review, Applicant can find no passages in *any* of the cited references which might suggest speculatively precharging a DRAM row based in any part on historical program execution data, as recited by independent claim 46. As such, Applicant respectfully traverses the Examiner's rejection.

In the present Application, speculative precharging is described in the context of an "intelligent caching" system. *See* Application, page 24, line 6 to page 25, line 20. Utilizing the intelligent cache methods described in the present Application ensures that data required by the functional units 128 is available when it is needed, thereby reducing or eliminating the additional wait time needed to assemble data when a cache miss occurs. *See id.* at page 25, lines 16-20; *see also* page 26, line 24 to page 27, line 6. As such, the overall speed of program execution is increased. *See id.* Furthermore, the present Application states that speculative precharging is predicted based at least partially on historical program execution data, such as a *branch history table* and/or *loop counters*. *See* Application, page 15, lines 4-7; page 30, line 29 to page 31, line 4; page 31, lines 23-29. By analyzing the historical program execution data, the system can make a *prediction* on which rows should be precharged based on *past program execution behavior*.

In rejecting independent claim 46, the Examiner admits that the Inagami and Luk references fail to disclose speculative precharging, but attempts to fill the deficiencies of Inagami and Luk using the Farmwald reference. Specifically, the Examiner stated:

> Inagami in view of Luk has not explicitly taught speculatively precharging (activating) a row pointed to by said at least one row register. However, it is known that precharging must occur. Farmwald has taught an instruction which explicitly speculatively precharges a row. See column 7, lines 49-55, and note that a bus transaction instruction is issued. It includes the fields shown in Fig. 4. When bit 2 of AccessType is set to the appropriate value or when the 3rd bit of AccessType is set to the appropriate value, a row precharge will occur. See column 12, lines 34-53, and column 11, line 25-60. Note that such a command allows for precharging ahead of time so that when an actual load occurs, for instance, the load can begin immediately, thereby saving time.
>
> Office Action, page 12. (Emphasis added).

However, after careful review, Applicant does not believe that Farmwald discloses speculatively precharging a DRAM row based in *any* part on historical program execution data, as recited by amended independent claim 46. Referring now to the specific passages cited by the Examiner, it appears that to the extent that the Farmwald reference discusses what the Examiner asserts is "speculative precharging," such precharging operations are determined based on program access modes, and *not* historical program execution data. *See* Farmwald, col. 11, lines 25-60. For example, Farmwald suggests that a DRAM processor may be configured to precharge a row when program access occurs in normal mode and to hold data (without precharging again) when program access occurs in page mode. *See id.* However, nothing in the Farmwald reference suggests that precharging the row based on the program access mode is based in any part on program execution history, as generally recited by claim 46.

The Farmwald reference also seems to suggest that the DRAM may be configured to precharge sense amps that have not been accessed for a selected period of time. *See id.* at lines 59-60. Even if monitoring the amount of idle time associated with the sense amps could be considered "historical data," issuing a precharge command based on the idle time does not take *past program execution data* into account. For example, just because certain sense amps have not been accessed for a certain period of time, it does not necessarily mean they will need to be accessed by the program in the near future. At very best, the methods described in Farmwald appear to be hard-coded procedures designed to precharge the DRAM when a certain condition occurs, regardless of past program execution behavior. To the contrary, Applicant notes that the present Application *clearly* states that speculative precharging is performed based at least partially upon *past program execution data*, for example, analyzing *branch history tables* for tracking the number of times a program has taken a certain branch in previous executions, as well as monitoring *loop counters* to determine how likely it is that the same execution path will occur. *See* Application, page 15, lines 4-7; page 30, line 29 to page 31, line 4; page 31, lines 23-29.

With this distinction in mind, Applicant asserts that none of the methods for precharging described by the Farmwald reference perform precharge operations based on historical program execution data. As such, Applicant respectfully asserts that Inagami and Farmwald, taken alone or in hypothetical combination, fail to disclose speculatively precharging a row based at least partially on *historical program execution data*, as recited by independent claim 46. Moreover, because none of the additional references cited by the Examiner, either alone or in combination, obviate the deficiencies of the Inagami, Luk and Farmwald references, Applicant respectfully asserts that the Examiner has failed to establish a *prima facie* case of obviousness with regard to this subject matter. For at least this reason, Applicant respectfully requests withdrawal of the rejection under 35 U.S.C. § 103(a) of independent claim 46, as well as those claims which depend thereon.

Deficiency #4: The cited references, alone or in combination, do not teach or suggest a command to deactivate a row pointed to by a row address register after the row has been precharged.

With regard to the fourth deficiency of the cited references noted above, Applicant notes that this subject matter is recited at least by independent claim 1. Specifically, amended independent claim 1 recites, generally, an instruction set including a command for *deactivating* a row pointed to by a row address register after the row has been precharged. In rejecting claim 1, the Examiner admits that the Inagami reference fails to disclose such a feature, but attempts to fill the deficiency of Inagami using the Farmwald reference. Specifically, the Examiner stated:

> Inagami ... has not explicitly taught a command to deactivate said row pointed to by said row address register after it had been precharged by the command to precharge. However, it is known that any load that occurs in a DRAM is a destructive load. That is, reading a value in DRAM causes the charges making up that value to diminish. Consequently, after a value is read, the DRAM row must be refreshed. Farmwald has also taught a command which performs an explicit refresh. See Fig. 4 and column 12, lines 58-60.
>
> Office Action, page 5. (Emphasis added).

It appears that the Examiner is attempting to equate a DRAM refresh command with the deactivate command recited by independent claim 1. Applicant respectfully traverses the Examiner's rejection.

In reaching the present rejection, Applicant asserts that the Examiner has misapplied long standing legal precedent stating that the pending claims must not only be given an interpretation that is reasonable and consistent with the *specification*, but also be consistent with the interpretation that *one of ordinary skill in the art* would reach. *See In re Prater*, 415 F.2d 1393, 1404-05, 162 U.S.P.Q. 541, 550-51 (C.C.P.A. 1969) (emphasis added); *see also In re Morris,* 127 F.3d 1048, 1054-55, 44 U.S.P.Q.2d 1023, 1027-28 (Fed. Cir. 1997); *see also In re Cortright*, 165 F.3d 1353, 1359, 49 U.S.P.Q.2d 1464, 1468 (Fed. Cir. 1999); M.P.E.P. § 2111. Further, Applicant respectfully notes that the Federal Circuit, sitting *en banc*, recently provided a summary and additional guidance regarding the proper interpretation of claims in view of the specification. *See Phillips v. AWH Corp.*, 75 U.S.P.Q.2d 1321 (Fed. Cir. 2005) (*en banc*). In *Phillips*, the Federal Circuit again emphasized the primacy of the specification in claim interpretation. Particularly, the *Phillips* court noted that the specification "is always highly relevant to the claim construction analysis. Usually, it is dispositive; *it is the single best guide to the meaning of a disputed term*." *Phillips*, 75 U.S.P.Q.2d at 1327 (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)) (Emphasis added). Moreover, the court also noted that:

> Ultimately, the interpretation to be given a term can only be determined and confirmed with a full understanding of what the inventors actually invented and intended to envelop with the claim. The construction that stays true to the claim language *and most naturally aligns with the patent's description of the invention* will be, in the end, the correct construction.

Phillips, 75 U.S.P.Q.2d at 1328-29 (quoting *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998)) (Emphasis added).

With the foregoing and controlling case law in mind, Applicant notes that the present Application *clearly* discloses that an embedded DRAM processor, in accordance with an embodiment of the present invention, has an instruction set including *both* a command to *activate* (precharge) a row and a command to *deactivate* a row in a DRAM array. *See* Application, page 8, lines 22-27; page 11, lines 22-28; page 21, lines 3-5. The commands are utilized by a data assembly unit 122 which is operatively configured to control the movement of data between register files. *See id.* at page 21, lines 3-5. Furthermore, the activate and deactivate commands are based on a set of *activation bits* 204 which may be set or cleared under program control to activate or deactivate entire rows of *selected* DRAM banks in the DRAM arrays 102. *See id.* at page 26, lines 5-10; page 31, lines 13-20; Fig. 3. The data assembly unit 122 assembles data corresponding to possible program branches into inactive register files so that the data required for each branch will be ready regardless of which branch a program ultimately takes. *See id.* at page 26, line 24 to page 27, line 6. A row address unit 306 together with the activation bits 304 (i.e., for activating and deactivating rows) are used together with masked row-oriented load/store commands for *controlling the movement of data* between the DRAM array and an inactive register file. *See id.* at page 32, line 29 to page 33, line 3. As such, the wait time required to assemble data when a cache miss occurs is preempted. *See id.* at page 26, line 24 to page 27, line

6. Indeed, it is *clear* from the specification that a deactivate command is associated with controlling the movement of data (i.e., via data assembly unit 122) between register files and a DRAM array, and is issued based on a set of activation bits manipulated under program control.

In contrast, a "refresh command", which the Examiner is attempting to equate with the deactivate command of independent claim 1, is *not* associated with the movement of data. As will be appreciated by those skilled in the art, DRAM is a type of random access memory which stores data into capacitors within an integrated circuit. However, since in real world operation, capacitors leak charge over time, the information stored within the capacitors eventually fades. Accordingly, the capacitors (memory cells) of a DRAM array must be refreshed periodically in order to maintain the data stored within the cells. A refresh is typically accomplished by reading the contents of a cell and then writing it back into the *same memory cell*, thus replenishing the charge stored in the refreshed cell.

In the present rejection, the Examiner's reasoning for equating a DRAM refresh operation to the "deactivate command" recited by claim 1 is that because during a refresh operation, the memory cell or cells being refreshed are inaccessible. *See* Office Action, page 5. In support of this position, the Examiner cites the Garcia reference, which discloses that during a refresh cycle, DRAM memory cells may unable to respond to read or write command requested by a scan test. *See* Garcia, col. 1, line 66 – col. 2, line 2. In other words, the Examiner seems to be suggesting that DRAM rows are "deactivated" during a refresh cycle because read or write operations cannot occur. However, Applicant asserts that the Examiner's logic is flawed. Rather than rendering the DRAM *unresponsive* to read or write operations, the refresh cycle itself is defined as including *both* a read *and* a write operation. To the extent that the Examiner's reliance on the Garcia or Farmwald references has any basis whatsoever, it is that the refresh cycle must complete its *own* read and write operations (i.e., reading data from a memory cell and writing it back into the memory cell) before *subsequent* read or write operations may be performed. It certainly does not mean the DRAM is "deactivated." Furthermore, nothing in the Farmwald or Garcia references suggest that the refresh command is associated with controlling the movement of data. Rather, as discussed above, a refresh command *keeps* data in its current location (i.e., reading data from a memory cell and writing it back into the same memory cell). As such, Applicant asserts that when read in view of the specification, one of ordinary skill in the art *would not* interpret a deactivate command to be the equivalent of a DRAM refresh command as the Examiner has so suggested.

Therefore, even if the Farmwald or Garcia references *do* suggest the use of a refresh command for refreshing one or more rows/cells of a DRAM array, a refresh command is simply *not the equivalent* of the deactivate command recited by independent claim 1 when interpreted in view of the specification. Accordingly, Applicant respectfully asserts that the Inagami, Farmwald, and Garcia references, taken alone or in hypothetical combination, fail to disclose a deactivate command as recited by claim 1. Further, because none of the additional references cited by the Examiner, either alone or in combination, obviate the deficiencies of the Inagami, Farmwald, or Garcia references, Applicant respectfully asserts that the Examiner has failed to establish a *prima facie* case of obviousness with regard to this subject matter. For at least this reason, Applicant respectfully requests withdrawal of the rejection under 35 U.S.C. § 103(a) of independent claim 1, as well as those claims which depend thereon.

Deficiency #5: The cited references, alone or in combination, do not teach or suggest register files capable of being placed into an active and an inactive state, nor does the prior art teach or suggest commands for manipulating active and inactive register files.

With regard to the fifth deficiency of the cited references noted above, Applicant notes that this subject matter can be found at least in amended independent claims 16, 28, 45, 49, and 50. For example, independent claims 16, 28, 45, 49, and 50, recite, generally, register files capable of being placed in an active and inactive state. Also, independent claims 28 and 45 recite, generally, commands for (1) placing an inactive register into an active state, and (2) for transferring data between a DRAM array and an inactive register file.

In the present Application, the active and inactive register files are described in the context of an intelligent cache structure that splits the embedded-DRAM architecture into a core execution portion and an intelligent data assembly portion. *See* Application, page 6, lines 6-21; page 24, line 6 to page 25, line 3. In order to implement intelligent caching, a data assembly unit 122 assembles data in inactive register files. *See id.* at page 6, lines 13-16. Assembling the data may involve performing register-to-register move operations, as well as load/store operations. *See id.* at page 24, lines 27-29. Once the data is *finished being assembled* by the data assembly unit 122, the register file switches from an inactive to an active state. *See id.* at page 6, lines 13-16. Once a register is switched to an active state, it functions as an architectural register, being visible to the program executed by functional units. *See id.* Indeed, as *clearly* set forth in the specification, active and inactive states are determined based on whether or not the data assembly has finished assembling data. Further, the intelligent caching scheme described in the present Application is an improvement over the prior art hit-or-miss caching schemes. *See id.* at page 25, lines 16-20.

Keeping the previously discussed legal precedent regarding claim interpretation in mind, Applicant reminds the Examiner that pending claims must be given an interpretation that is reasonable and consistent with the *specification.* See *In re Prater*, 415 F.2d 1393, 1404-05, 162 U.S.P.Q. 541, 550-51 (C.C.P.A. 1969) (Emphasis added); *see also In re Morris,* 127 F.3d 1048, 1054-55, 44 U.S.P.Q.2d 1023, 1027-28 (Fed. Cir. 1997). In rejecting the pending claims, the Examiner admits that the Inagami reference fails to disclose that the vector registers (VR0-VR7) are capable of being placed in an active or inactive state. However, the Examiner attempts to fill this deficiency using the Parady reference, which generally describes a method for program thread switching. Specifically, the Examiner asserted that Parady teaches a switch command for toggling between active and inactive states based on which *program thread* is currently active. *See* Office Action, pages 26, 31, 41, 44, 47. However, Applicant asserts that this interpretation of the terms "active" and "inactive" is not reasonable and consistent with the Applicant's specification, as discussed above, which *clearly* states that "active" or "inactive" register files are determined based on whether or not a data assembly unit has *completed assembly of data* within the register file. Furthermore, as discussed above, the present Application describes intelligent caching, which is an *improvement* over the prior art hit-or-miss cache scheme, whereas the method taught by Parady relies on the prior art hit-or-miss scheme. For example, the Parady reference discloses that "[t]he indication that a thread switch is required is provided on a line 114 providing an L2 miss indication from cache control/system interface 22." Parady, col. 3, lines 56-65. (Emphasis added). Indeed, the Parady reference clearly shows that toggling between register files is based on a thread switch, not the *assembling of data* within a register file. As such, Applicant asserts that the Examiner's interpretation of the terms "active" and "inactive" is not reasonable and consistent with the specification. Therefore, Applicant asserts that neither the Inagami nor the Parady references teach or even suggest that register files can be placed into active and inactive states.

Additionally, as noted above, the Examiner further rejected claims 28 and 45, stating that the Inagami reference, in combination with the Parady reference, discloses a command for placing an inactive register into an active state. However, in view of the foregoing discussion regarding the interpretation of the terms "active" and "inactive," Applicant believes it is clear that Parady does not teach transitioning register files between "active" and "inactive" states as described in the context of the present Application (i.e., based on assembling of data by a data assembly unit) and, therefore, the Parady reference does not teach a command for doing the same.

The Examiner also rejected claims 28 and 45, stating that the Inagami reference, in combination with Parady, and in further view of the Bissett reference, discloses a command for storing data from a DRAM array into an inactive register file. However, as discussed above, the Parady reference fails to disclose active and inactive registers. Furthermore, it does not appear that the Bissett reference obviates the deficiencies of either the Inagami or Parady references. The Bissett reference merely discloses the general concept that certain operations can be performed without influencing processor operation. *See* Bissett, col. 4, lines 3-9. Moreover, it does not appear that Bissett discloses active and inactive register files, as recited in the pending claims.

In view of the foregoing analysis, Applicant respectfully asserts that the Inagami, Parady, and Bissett references fail to disclose the following: (1) register files capable of being places in an active and inactive state; (2) a command for placing a register into an active or inactive state; and (3) a command to transfer data between a row of a DRAM array and an inactive register. Moreover, because none of the additional references cited by the Examiner, either alone or in combination, obviate the deficiencies of the Inagami, Parady, or Bissett references, Applicant respectfully asserts that the Examiner has failed to establish a *prima facie* case of obviousness with regard to this subject matter. For at least this reason, Applicant respectfully requests withdrawal of the rejection under 35 U.S.C. § 103(a) of independent claims 16, 28, 45, 49, and 50, as well as those claims which depend thereon.

Request Allowance of All Pending Claims

In view of the remarks presented in this Response, it is believed Applicant has clearly demonstrated that with regard to the subject matter recited in at least the independent claims, any possible combination of the Inagami, Parady, Luk, Farmwald, and Bissett references fail to teach or suggest at least one of the above discussed claim features. Further, due to the length and complexity of both the instant Office Action and the present Response, Applicant has summarized, for the Examiner's convenience, the deficiencies of the cited references with regard to each independent claim discussed herein, as follows:

Independent claim 1

With regard to independent claim 1, the cited references, alone or in combination, do not disclose (1) a command for deactivating a row of a DRAM array, or (2) register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal or in a single operation.

Independent claim 13

With regard to independent claim 13, the cited references, alone or in combination, do not disclose register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal or in a single operation.

Independent claim 16

Further, with regard to independent claim 16, the cited references, alone or in combination, do not disclose (1) register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal or in a single operation, or (2) register files capable of being placed in an active or inactive state.

Independent claim 23

Further, with regard to independent claim 23, the cited references, alone or in combination, do not disclose (1) a dual port register file having a first access port operative to transfer data between a register file *and* a DRAM, and a second access port operative to transfer data between a register file and a functional unit, or (2) register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal or in a single operation.

Independent claim 28

Further, with regard to independent claim 28, the cited references, alone or in combination, do not disclose (1) register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal or in a single operation, or (2) register files capable of being placed in an active or inactive state.

Independent claim 45

Further, with regard to independent claim 45, the cited references, alone or in combination, do not disclose (1) a dual port register file having a first access port operative to transfer data between a register file *and* a DRAM, and a second access port operative to transfer data between a register file and a functional unit, or (2) register files capable of being placed in an active or inactive state.

Independent claim 46

Further, with regard to independent claim 46, the cited references, alone or in combination, do not disclose (1) register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal or in a single operation, or (2) speculatively precharging DRAM based at least partially on historical program execution data.

Independent claim 49

Additionally, with regard to independent claim 49, the cited references, alone or in combination, do not disclose (1) register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal or in a single operation, or (2) register files capable of being placed in an active or inactive state.

Independent claim 50

Finally, with regard to independent claim 50, the cited references, alone or in combination, do not disclose (1) register files capable of loading or storing an entire row of a DRAM array in response to a single latch signal or in a single operation, or (2) a dual port register file having a first access port operative to transfer data between a register file *and* a DRAM, and a second access port operative to transfer data between a register file and a functional unit.

Accordingly, because Applicant has clearly shown that the cited references, alone or in combination, do not disclose *every* feature recited in the independent claims, the Examiner has not established a *prima facie* case of obviousness at least against independent claims 1, 13, 16, 23, 28, 45, 46, 49, and 50. Therefore, Applicant respectfully request withdrawal of the rejections under 35 U.S.C. § 103(a), and allowance of claims 1, 13, 16, 23, 28, 45, 46, 49, and 50, as well as any claims dependent thereon.

General Authorization for Payment of Fees and Extensions of Time

Applicant does not believe that any fees are due; however, if any fees are due at this time or during the pendency of this application, the Commissioner is authorized to charge such requisite fees to Deposit Account No. 06-1315; Order No. MICS:0171-2/MAN. Furthermore, in accordance with 37 C.F.R. § 1.136, Applicant hereby provides a general authorization to treat this and any future reply requiring an extension of time as incorporating a request therefor and authorizes the Commissioner to charge any fees associated therewith to the above-referenced Deposit Account.

Conclusion

In view of the remarks and amendments set forth above, Applicant respectfully requests allowance of the pending claims. If the Examiner believes that a telephonic interview will help speed this application toward issuance, the Examiner is invited to contact the undersigned at the telephone number listed below.

Respectfully submitted,

Date: November 21, 2007

/Robert A. Manware/
Robert A. Manware
Reg. No. 48,758
FLETCHER YODER
7915 FM 1960 West, Suite 330
Houston, TX 77070
(281) 970-4545